A Hybrid R-BILSTM-C Neural Network Based Text Steganalysis

Yan Niu, Juan Wen, Ping Zhong, and Yiming Xue, *Member, IEEE,*

***Abstract*—With the emergence of the generation-based steganography, the traditional text steganalysis methods show the unsatisfactory detection performance as the manually extracted features are simple and non-universal. The recently proposed deep learning-based text steganalysis methods can obtain the great detection accuracy by extracting the high-level features. In this paper, a hybrid text steganalysis method (R-BILSTM-C) is proposed through combining the advantages of Bidirectional Long Short Term Memory Recurrent Neural Network (Bi-LSTM) and Convolutional Neural Network (CNN). The proposed method can efficiently capture both local features and long-term semantic information from text to improve the detection accuracy. In the proposed method, the Bi-LSTM architecture is used to capture the long-term semantic information of texts. And the asymmetric convolution kernels with different sizes are applied to extract the local relationship between words. In addition, the high dimensional semantic feature space is visualized. Experimental results show that the proposed method adapts to the different steganographic algorithms efficiently, and achieves the comparable or superior detection performance for the various sentence lengths compared with other state-of-the-art text steganalysis methods.**

***Index Terms*—Text steganalysis, Bi-LSTM, CNN, long-term semantic feature, local feature.**

I. Introduction

LINGUISTIC steganography that embeds the secret information into texts has attracted widespread attention as the most frequently used texts in daily life can provide a large number of carriers for text steganography. Generally, the linguistic steganography can be roughly divided into two main sorts: embedded-steganographic algorithms [1]–[3] and generation-based steganographic algorithms [4]–[6]. In the embedded-steganographic algorithms, the synonym substitution based steganography is widely used as it is hardly to cause the semantic changes after substitution. The generation-based steganography utilizes the powerful feature extraction and expression abilities of neural networks to acquire statistical and semantic features of the large number of training samples, and then generates the high-quality steganographic texts.

As the counter-technique of steganography, text steganalysis that aims to detect the existence of secret messages in the text has been rapidly developed. Most of the traditional text steganalysis methods are proposed based on the general machine learning framework [7]–[14]. However, these traditional steganalysis methods are difficult to adapt to the different kinds of steganographic algorithms since they are designed based on the statistical changes caused by a specific steganography. And they show the unsatisfactory detection performance for the latest generation-based text steganographic algorithms as the manually extracted features, such as word frequency distribution [8]–[11], and context fitness [10], are simple and non-universal. With the development of the generation-based text steganography [4]–[6], some researchers have studied the text steganalysis algorithms based on deep learning [15]–[17]. Wen *et al*. [15] propose a text steganalysis model to capture the local correlations between words based on CNN. Yang *et al*. [16] utilize the strong feature expression capability of the Recurrent Neural Networks(RNNs) to extract the long-term semantic features. Although the current deep learning-based steganalysis methods have achieved the great detection performance for distinguishing the stego texts through extracting the high-level features, they can be still improved. Notice that CNN is able to capture local semantic correlations of texts but it does not perform well in learning long-term sequential information, while RNN is ideal for processing sequences of any length [18]. And the Long Short Term Memory (LSTM), as a variant of RNN, is able to capture long-term contextual dependency and solve the problem of the vanishing gradient of the RNN.

In this paper, we propose a hybrid and universal text steganalysis algorithm based on deep learning, named R-BILSTM-C, to extract the local and global features by combining Bi-LSTM with CNN. The proposed text steganalysis scheme finds out the subtle differences in semantic spatial distribution before and after embedding the secret messages. It converts each sentence into the corresponding matrix by the fusion strategy in the word embedding layer firstly, and then concatenates the forward semantic features and back semantic information by Bi-LSTM to better express the long-term contextual features and the word order information. Inspired by Inception modules in [19], we employ the asymmetric convolution kernels with different sizes to extract the local features, which can not only improve the performance of the model, but also accelerate the training process and relieve over-fitting by reducing a large number of parameters. Thus, it can be concluded that the proposed method can effectively detect the existence of hidden information by extracting rich features which include not only the local relationship between words, but also the long-term semantic information. In addition, in order to solve the problem of the loss of

This work was supported by the National Natural Science Foundation of China under Grant 61872368 and Grant 61802410.

Y. Niu, J. Wen and Y. Xue are with the College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China(e-mail: niuyan@cau.edu.cn; wenjuan@cau.edu.cn; xueym@cau.edu.cn).

P. Zhong is with the College of Science, China Agricultural University, Beijing 100083, China (e-mail: zping@cau.edu.cn) (Corresponding author: Ping Zhong)

information caused by too many convolutional operations, the residual shortcuts block is used in the proposed method. Finally, the experimental results demonstrate that the proposed text steganalysis method can effectively detect stego texts with different sentence lengths and achieve the comparable or superior performance compared with other state-of-the-art text steganalysis methods.

The rest of the article is organized as follows: Section II introduces a brief overview of text steganalysis. A detail explanations of the proposed steganalysis method are elaborated in Section III. Section IV presents the experimental settings and analyzes the experimental results in detail. Finally, conclusions are drown in Section V.

II. THE PROPOSED METHOD

Benefitting from the relevant studies on the use of CNN and Bi-LSTM [20], [21], we propose a novel text steganalysis method which is mainly composed of the word-embedding layer, the Bi-LSTM layer and the CNN layer. The overall architecture is shown in Fig.1.

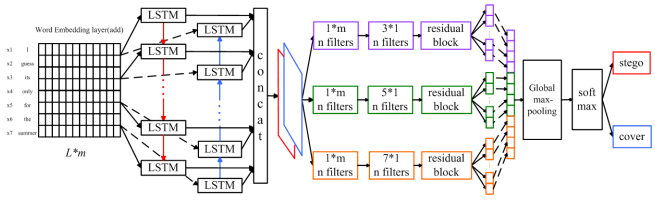


Fig. 1. The structure of the proposed steganalysis model

In the proposed method, the input sentence with l words is represented as:

$$S = \{x_1, x_2, \ldots, x_l\} \tag{1}$$

where x_i indicates the i-th word in sentence S, and l represents the length of sentence. In the word-embedding layer, the double-channel strategy is adopted, and the word sequence of each input sentence S is mapped into two matrices $D_1, D_2 \in R^{l\times d}$. Each word is represented as a d-dimension vector. The matrix D_1 is initialized by the publicly available word2vec vectors pre-trained on 100 billion words from Google News. Another matrix D_2 is initialized by randomly sampling from the uniform distribution in [-1,1]. The output matrix $D \in R^{d\times l}$ of the word embedding layer is obtained by adding D_1 and D_2.

The LSTM has been widely used in the text classification filed as it can not only deal with the features of sequential signals effectively, but also overcome the problems of gradient disappearance and gradient explosion. However, the traditional single-direction LSTM models can only be propagated in forward. In order to make every moment contain the context information, the Bi-LSTM layer is introduced in the proposed model. The contextual semantics of each sentence has two expressions, including forward and backward semantic information. The expressions are shown as follows:

$$\overrightarrow{h_t}, \overrightarrow{c_t} = \overrightarrow{LSTM}(x_t, \overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}), t \in \{1, \cdots, l\} \tag{2}$$

$$\overleftarrow{h_{l-t+1}}, \overleftarrow{c_{l-t+1}} = \overleftarrow{LSTM}(x_t, \overleftarrow{h_{l-t}}, \overleftarrow{c_{l-t}}), t \in \{l, \cdots, 1\} \tag{3}$$

where $x_t \in R^{d\times 1}$ is the input at time t, and it is also the word vector corresponding to the t-th word in the sentence. h_t,$c_t \in R^{1\times d}$ are the hidden state and memory state at time t, respectively. The overall representation can be acquired as $H = [\overrightarrow{H_f}, \overleftarrow{H_b}] \in R^{l\times d\times 2}$, where $\overrightarrow{H_f} = (\overrightarrow{h_1}, \cdots, \overrightarrow{h_l})$ is the forward hidden state matrix and $\overleftarrow{H_b} = (\overleftarrow{h_l}, \cdots, \overleftarrow{h_1})$ is the backward hidden state matrix. In the CNN layer, the asymmetric convolution kernels with different sizes are used to acquire the local correlations between words. The asymmetric convolution kernels are filter vectors which are used to slide over the sequence and capture features at different positions. In other word, we factorize the traditional filters with the size of $d \times k$ into two filter vectors with smaller sizes of $d \times 1$ and $1 \times k$, respectively. The feature maps I are extracted by the first convolutional operation, and the expression as following:

$$I = f_{relu}(W_j^1 \cdot H + b) \tag{4}$$

where $j \in \{1, \cdots, n\}$ is used to mark the channels of the feature map and n is the number of filters. ${W_j}^1 \in R^{1\times d\times 2}$ is the weight of the first convolution kernel, and H ($H \in R^{l\times d\times 2}$) is the output of the Bi-LSTM layer. f_{relu} is the Relu function which can increase the nonlinearity of the feature.

The weight of filter in the second convolution operation is defined as $W_u^2 \in R^{k\times 1\times n}$. The corresponding feature maps are

$$F_u = f_{relu}(W_u^2 \cdot I + b) \tag{5}$$

where $u \in \{1, \cdots, n\}$ is an index of feature map extracted by the second convolution operation. All of F_u form a tensor $F \in R^{1\times l\times n}$. To further choose the proper kernel sizes and verify the effectiveness of the asymmetric convolution kernels, we have experimented with different filters, and the results are shown as Fig.2. According to the results, the asymmetric convolution kernels of sizes $\{3, 5, 7\}$ are selected to extract the richer features, and three tensors with the same shape are concatenated as a final feature map $F^* \in R^{1\times l\times 3n}$. In addition,

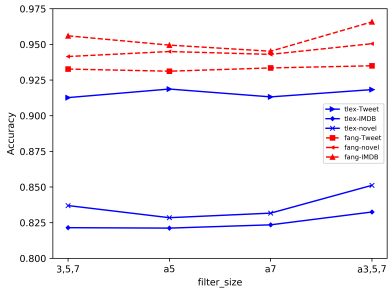


Fig. 2. The performance of detecting stego texts with different kernel sizes by R-BILSTM-C, where ‘a5’ and ‘a7’ represent the asymmetric convolution kernels with the sizes of 5 and 7, respectively, and ‘a3,5,7’ represent the asymmetric convolution kernels with the sizes of 3,5,7.

the residual shortcuts block is used in the model to solve the problem of information loss caused by too many convolutional operations. We have carried out a series of experiments to verify the effectiveness of the residual shortcut block. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2019.2953953, IEEE Signal Processing Letters

Table I. The efficiency of residual shortcuts block

steganalysis model	Tweet		Movie		Gutenberg	
	T-lex	Tina-Fang	T-lex	Tina-Fang	T-lex	Tina-Fang
model with residual block	0.919	0.935	0.833	0.989	0.851	0.951
model without residual block	0.910	0.932	0.828	0.978	0.845	0.959

results are show as Table I. It indicates that the residual block can effectively improve the detection performance.

Next, the global max pooling layer is applied to reduce dimension and select the most important features. The last part of the proposed model is fully connected layer, where soft-max function is used as classifier to discriminate stego texts from normal texts.

III. EXPERIMENTS AND ANALYSIS

A. Experimental Setting and Training Details

In the experiments, two text steganography algorithms are detected: T-Lex [1], and Tina-Fang [4]. T-Lex is a typical embedded-steganographic algorithm which embeds the secret messages by synonym substitution. The another is a generated-based steganographic algorithm which generates stego text by LSTM network. Three kinds of text carries are used in the experiments, including Twitter [22], IMDB [23] and Gutenberg [24]. The details of three datasets are described in Table II.

Table II. Datasets information

Data	number of sentence				average sentence length	max sentence length
	training set		testing set			
	cover	stego	cover	stego		
Tweet	8000	8000	2000	2000	8	48
IMDB	8000	8000	2000	2000	17	459
Gutenberg	5000	5000	1000	1000	24	173

Some hyper-parameters are set as follows: the dimensionality of hidden state in Bi-LSTM layer is 300. It equals to the dimension of word embeddings layer. The batch size is set to 64. As for the regularization, the dropout rates in word-embedding layer [25], Bi-LSTM layer and penultimate layer are 0.5, 0.2, 0.4, respectively. The number of filters is 128. Besides, the mini-batch gradient descent with Adam algorithm is used to train the proposed model, and the initial learning rate is 0.001. In order to reflect the performance of the proposed model, three representative text steganalysis algorithms are chosen for comparison, including synonym substitution-based steganalysis(SS) [11], LS-CNN [15], and TS-BiRNN [16]. In the experiments, the Accuracy, Recall, Precision are used to evaluate the performance of these text steganlysis methods.

B. Results

1) Detection Performance Comparison For Different Steganographic Methods : The experimental results for the different stego datasets are shown in Table III. Firstly, in most steganographic datasets, the proposed text steganalysis method shows the best detection performance regardless of the data formats, embedding rates and steganographic algorithms. It indicates that the proposed text steganalysis method can capture the subtle semantic changes of the stego texts successfully and discriminate the stego texts from normal texts effectively. Secondly, the different embedding rates are used for Fang steganographic algorithm. It can be observed that the detection performance of each steganalysis method is improved with the increase of the embedding rate. It is concluded that the higher the embedding rate is, the more seriously the semantic relationship between words is destroyed, and the easier it is for the steganalysis methods to detect the hidden information.

In addition, in order to assess the effectiveness of the important blocks in the proposed model, a series of experiments are conducted. For the word-embedding layer, we compare three different learning strategies: the 'dynamic' mode, the 'static' mode, and the 'double-channel' mode. The 'static' mode uses matrix D_1 in the word-embedding layer to improve the generalization ability in the absence of a large supervised training set [26]. And the 'dynamic' mode learns word vector dynamically during the training process by using matrix D_2. We combine the above learning strategies as the 'double channel' mode. The compare results are shown in Fig.3. It is demonstrated that the 'double-channel' fusion strategy can improve the detection performance of the proposed text steganalysis.

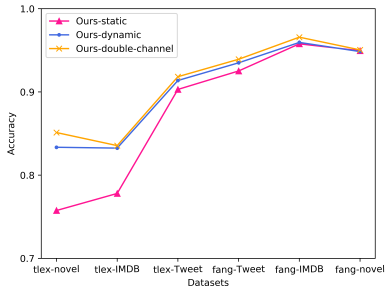


Fig. 3. The performance of detecting stego texts with different word-embedding strategies

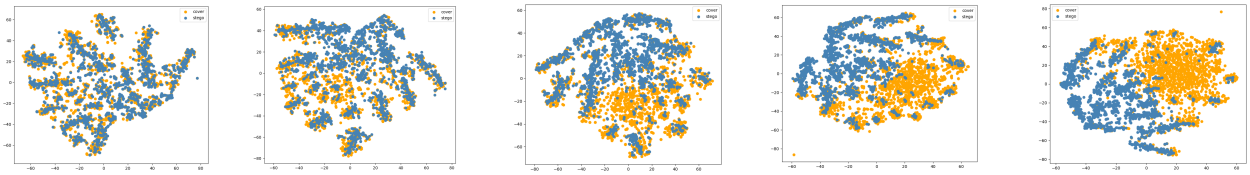
For the residual shortcuts blocks (He *et al*. [27], 2016) in the proposed text steganalysis method, we remove them from the model and observe the changes in detection accuracy. It can be seen that the detection accuracy of the model without the residual shortcuts blocks on the fang-tweet dataset has slightly worse than the model with these blocks. Besides, on the others datasets, the similar detection results are obtained. It can be proved that the residual shortcuts blocks can reduce the loss of information and improve the detection performance of the text steganalysis.

2) Detection Performance Comparison For Different Lengths Of Sentence : To further illustrate the effect of the sentence lengths on the detection results, we split stego texts into different levels according to the length of sentences: $0-40$, $40-80$, $80-120$, $120+$; The results are shown in Table IV. It can be firstly observed that the proposed steganalysis method has comparable or superior performance to the other steganalysis methods regardless of the lengths of sentence. In addition, for the generation-based steganography, the longer the sentence is, the more accurate of the semantic expression can be captured by text steganalysis, and the detection of stego texts will be more easier. However, for the embedded-steganographic algorithms, it seems that the detection accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2019.2953953, IEEE Signal Processing Letters

Table III. The detection results of different steganalysis methods

method		SS			LS-CNN			TS-BiRNN			OURS		
data	steganography	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Tweet	T-Lex	0.635	0.548	0.664	0.916	**0.954**	0.887	0.913	0.945	0.887	**0.919**	0.939	**0.901**
	Fang(bpw=1)	0.509	0.165	0.554	**0.749**	0.731	**0.759**	0.732	0.697	0.749	0.741	**0.737**	0.741
	Fang(bpw=2)	0.542	0.254	0.667	0.846	**0.905**	0.810	0.842	0.863	0.829	**0.852**	0.873	**0.837**
	Fang(bpw=3)	0.547	0.269	0.676	0.921	0.905	0.935	0.927	0.910	0.942	**0.935**	**0.916**	**0.952**
	Fang(bpw=4)	0.650	0.601	0.817	0.934	0.954	0.917	0.934	0.955	0.917	**0.940**	**0.959**	**0.918**
	Fang(bpw=5)	0.702	0.578	0.849	0.950	**0.981**	0.924	0.944	0.963	0.928	**0.952**	0.974	**0.934**
Movie	T-Lex	0.607	0.745	0.584	**0.835**	0.807	**0.834**	0.821	0.835	0.813	0.833	**0.870**	0.810
	Fang(bpw=1)	0.514	0.313	0.523	0.943	0.941	0.945	0.914	0.918	0.911	**0.957**	**0.952**	**0.961**
	Fang(bpw=2)	0.560	0.412	0.619	0.964	0.943	**0.984**	0.959	0.929	0.987	**0.966**	**0.963**	0.966
	Fang(bpw=3)	0.610	0.410	0.598	0.985	0.986	0.987	0.972	0.981	0.970	**0.989**	**0.988**	**0.989**
	Fang(bpw=4)	0.623	0.451	0.688	0.983	0.994	0.969	0.984	0.996	0.971	**0.994**	**0.997**	**0.971**
	Fang(bpw=5)	0.636	0.519	0.628	0.992	0.997	0.987	0.995	0.997	0.993	**0.998**	**0.998**	**0.998**
Gutenberg	T-Lex	0.751	0.677	0.795	0.848	0.795	0.888	0.846	0.883	0.822	**0.851**	**0.797**	**0.893**
	Fang(bpw=1)	0.528	0.316	0.548	0.814	**0.922**	0.758	0.808	0.898	0.761	**0.849**	0.890	**0.824**
	Fang(bpw=2)	0.590	0.421	0.635	0.940	**0.948**	0.933	0.924	0.923	0.924	**0.942**	0.946	**0.936**
	Fang(bpw=3)	0.613	0.467	0.659	0.945	0.948	0.943	0.936	0.943	0.929	**0.951**	**0.955**	**0.946**
	Fang(bpw=4)	0.622	0.507	0.658	0.955	**0.975**	0.945	0.945	0.955	0.954	**0.961**	0.973	**0.950**
	Fang(bpw=5)	0.744	0.726	0.753	0.974	0.998	0.965	0.975	0.996	0.956	**0.977**	**0.999**	**0.967**



(a) 1 bit/word. (b) 2 bit/word. (c) 3 bit/word. (d) 4 bit/word. (e) 5 bit/word.

Fig. 4. The visualization of feature space distribution for Tweet dataset

Table IV. The detection results for different lengths of sentence

Data	Steganography	Length of sentence	SS	LS-CNN	TS-BiRNN	Our method
Twitter	T-Lex	0-40	0.636	0.934	**0.945**	0.935
		40-80	0.639	0.917	0.895	**0.925**
		80-120	0.623	0.907	0.905	**0.913**
		120+	0.620	0.899	0.875	**0.901**
	Tina-Fang	0-40	0.540	0.870	0.875	**0.910**
		40-80	0.540	0.980	0.975	**0.990**
		80-120	0.600	0.985	0.985	**1.000**
		120+	0.610	**1.000**	**1.000**	**1.000**
IMDB	T-Lex	0-40	0.627	0.960	0.950	**0.970**
		40-80	0.599	0.930	0.935	**0.950**
		80-120	0.686	0.905	**0.925**	**0.925**
		120+	0.608	**0.895**	0.885	**0.895**
	Tina-Fang	0-40	0.500	0.980	0.985	**0.990**
		40-80	0.550	0.990	**0.995**	**0.995**
		80-120	0.535	0.992	**0.995**	**0.995**
		120+	0.585	0.995	0.997	**1.000**
Gutenberg	T-Lex	0-40	0.636	0.871	0.850	**0.893**
		40-80	0.740	0.820	0.840	**0.845**
		80-120	0.780	**0.815**	0.790	0.810
		120+	0.745	**0.849**	0.825	0.830
	Tina-Fang	0-40	0.470	0.835	0.795	**0.860**
		40-80	0.485	0.890	0.895	**0.925**
		80-120	0.530	**0.950**	0.920	**0.950**
		120+	0.520	**0.980**	**0.980**	**0.980**

decreases as the length increases. The reasons for the phenomenon maybe that the semantic distribution spaces between cover and stego are similar. As the sentence becomes longer, the difference becomes negligible in the semantic spaces, which results in extremely similar features being extracted. So the detection accuracy is reduced.

3) Visualization analysis: We reduce the dimensionality and visualize feature space by the t-Distributed Stochastic Neighbor Embedding (t-SNE) [28] technique. In the feature space, each point represents a sentence. The results on Tweet dataset are shown in Fig.4, and the performances on the other datasets are similar. It can be obtained from Fig.4, with the embedded rate of hidden information increasing, the distribution of the semantic space has changed obviously. In the semantic space, the blue points which represent the stego sentences generated by Tina-Fang's model [4] gradually spread and deviate from the distribution of the yellow poits which represent the cover sentences, and the area formed by the blue points has clear boundaries with the yellow dot area.

IV. Conclusion

In this paper, we propose an efficient and hybrid text steganalysis method based on CNN network and Bi-LSTM network. The architecture distinguishes the stego texts from the normal texts by extracting long-term and local features. The long-term semantic information which incorporates the forward and backward semantic information can be extracted by the Bi-LSTM layer. Instead of the normal filters, the multiple asymmetric convolutional kernels with different sizes are used in CNN layer to extract the local relationship between words. This joint strategy makes the proposed text steganalysis method excellently capture the subtle changes in the semantic space before and after embedding the secret information. To further show the effectiveness of the text steganalysis algorithm, we have made a visual analysis of the changes in the semantic space by the t-SNE technique. Extensive experiments have demonstrated the excellent performance of the proposed text steganalysis method over the state-of-the-art text steganalysis methods, regardless of the kinds of steganographic algorithms and the lengths of carrier sentences.

REFERENCES

[1] K. Winstein, "Lexical steganography through adaptive modulation of the word choice hash," *Available: http://web.mit.edu/keithw/tlex/*, 1998.

[2] L.Huo and Y.Xiao, "Synonym substitution-based steganographic algorithm with vector distance of two-gram dependency collocations," in *Proc.IEEE Int.Conf.Comput.Commun.*, 2016, pp. 2776–2780.

[3] L. Xiang, W. Wu, X. Li, and C. Yang, "A linguistic steganography based on word indexing compression and candidate selection," *Multimedia Tools & Applications*, vol. 77, no. 21, pp. 1–21, 2018.

[4] T. Fang, M. Jaggi, and K. Argyraki, "Generating steganographic text with lstms," *arXiv preprint arXiv:1705.10742*, 2017.

[5] Y. Luo and Y. Huang, "Text steganography with high embedding rate: Using recurrent neural networks to generate chinese classic poetry," in *Acm Workshop on Information Hiding & Multimedia Security*, 2017, pp. 99–104.

[6] Z. Yang, X. Guo, Z. Chen, Y. Huang, and Y.-J. Zhang, "Rnn-stega: Linguistic steganography based on recurrent neural networks," *IEEE Transactions on Information Forensics and Security*, 2018.

[7] C. M. Taskiran, M. Topkara, and E. J. Delp, "Attacks on lexical natural language steganography systems," in *Proc.SPIE Int.Soc.Opt.Eng.*, vol. 37. International Society for Optics and Photonics, Nov.2011, pp. 1071–1081.

[8] Z. Chen, L. Huang, P. Meng, W. Yang, and H. Miao, "Blind linguistic steganalysis against translation based steganography," in *Proc.Int.Workshop.Digit.Watermarking*, Oct.2010, pp. 251–265.

[9] H. Yang and X. Cao, "Linguistic steganalysis based on meta features and immune mechanism," *Chinese Journal of Electronics*, vol. 19, no. 4, pp. 661–666, 2010.

[10] Z. Chen, L. Huang, H. Miao, W. Yang, and P. Meng, "Steganalysis against substitution-based linguistic steganography based on context clusters," *Computers & Electrical Engineering*, vol. 37, no. 6, pp. 1071–1081, 2011.

[11] L. Xiang, X. Sun, G. Luo, and B. Xia, "Linguistic steganalysis using the features derived from synonym frequency," *Multimedia tools and applications*, vol. 71, no. 3, pp. 1893–1911, 2014.

[12] R. Din and A. Samsudin, "Performance analysis on text steganalysis method using a computational intelligence approach," in *Proceeding of International Conference on Electrical Engineering, Computer Science and Informatics(EECSI 2015), Palembang, Indonesia*, 2015, pp. 19–20.

[13] S. Samanta, S. Dutta, and G. Sanyal, "A real time text steganalysis by using statistical method," in *IEEE International Conference on Engineering & Technology*, 2016, pp. 264–268.

[14] L. Xiang, J. Yu, C. Yang, D. Zeng, and X. Shen, "A word-embedding-based steganalysis method for linguistic steganography via synonym-substitution," *IEEE Access*, pp. 64 131–64 141, 2018.

[15] J. Wen, X. Zhou, P. Zhong, and Y. Xue, "Convolutional neural network based text steganalysis," *IEEE Signal Processing Letters*, vol. 26, no. 3, pp. 460–464, 2019.

[16] Z. Yang, K. Wang, J. Li, Y. Huang, and Y.-J. Zhang, "Ts-rnn: Text steganalysis based on recurrent neural networks," *IEEE Signal Processing Letters*, 2019.

[17] Z. Yang, Y. Huang, and Y. Zhang, "A fast and efficient text steganalysis method," *IEEE Signal Processing Letters*, vol. 26, no. 4, pp. 627–631, 2019.

[18] H. Yang, Z. Yang, and Y. Huang, "Steganalysis of voip streams with cnn-lstm network," in *IH & MMSec*, July 3-5 2019.

[19] S. Christian, V. Vincent, S. Ioffe, S. Jon, and W. Zbigniew, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] Y. Kim, "Convolutional neural networks for sentence classification," *Eprint Arxiv*, 2014.

[21] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Netw*, vol. 18, no. 5, pp. 602–610, 2005.

[22] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, 2009.

[23] A. L. Maas, R. E. Daly, P. T. Pham, H. Dan, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2011, pp. 142–150.

[24] S. Lahiri, "Complexity of word collocation networks: A preliminary structural analysis," *arXiv preprint arXiv:1310.5111*, 2013.

[25] D. Shen, G. Wang, and W. Wang, "Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms," *arXiv preprint arXiv:1805.09843*, 2018.

[26] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[28] L. V. D. Maaten, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, 2014.